



# Connect the Campus

**Link submit:**

[https://uva.onlinejudge.org/index.php?option=onlinejudge&page=show\\_problem&problem=1338](https://uva.onlinejudge.org/index.php?option=onlinejudge&page=show_problem&problem=1338)

**Solution:**

C++	<a href="http://ideone.com/BZ8q5n">http://ideone.com/BZ8q5n</a>
Java	<a href="https://ideone.com/6GMigh">https://ideone.com/6GMigh</a>
Python	<a href="https://ideone.com/os4go2">https://ideone.com/os4go2</a>

**Tóm tắt đề:**

Cho  $N$  tòa nhà nằm ở các tọa độ  $(x, y)$  trên bản đồ, mỗi tòa nhà ở một tọa độ khác nhau, không có hai tòa nhà cùng một tọa độ. Các tòa nhà được đánh số từ  $1 \rightarrow N$ . Cho danh sách  $M$  cặp tòa nhà đã có kết nối dây cáp với nhau. Nhiệm vụ của bạn là xác định cần ít nhất bao nhiêu đoạn dây cáp nữa để tất cả các tòa nhà đều kết nối với nhau.

**Input:**

Có nhiều bộ test cần xử lý cùng một lúc, mỗi bộ test chứa các thông tin sau:

- Dòng đầu tiên chứa số lượng tòa nhà  $N$  ( $1 \leq N \leq 750$ ).
- $N$  dòng tiếp theo mỗi dòng chứa tọa độ  $(x, y)$  là tọa độ của từng ngôi nhà giá trị không vượt quá 10000.
- Dòng tiếp theo chứa số  $M$  ( $0 \leq M \leq 1000$ ) là số lượng cặp tòa nhà đã kết nối sẵn.
- $M$  dòng tiếp theo là danh sách các cặp tòa nhà.

**Output:**

Với mỗi bộ test, in ra tổng chiều dài cáp mới bạn định sử dụng để tất cả các tòa nhà được kết nối với nhau. Kết quả lấy phía sau dấu chấm hai chữ số.

**Ví dụ:**

4 103 104 104 100 104 103 100 100 1 4 2	4.41 4.41
---	--------------

4	
103 104	
104 100	
104 103	
100 100	
1	
4 2	

### Giải thích ví dụ:

Ví dụ gồm hai bộ test. Trong bộ đầu tiên có 4 tòa nhà lần lượt nằm tại các tọa độ:

- Tòa nhà 1 (103, 104)
- Tòa nhà 2 (104, 100)
- Tòa nhà 3 (104, 103)
- Tòa nhà 4 (100, 100)

Chỉ có hai tòa nhà có kết nối là tòa nhà 4 và tòa nhà 2, các tòa nhà còn lại chưa có kết nối. Vì thế bạn phải tìm đường kết nối sao cho kết quả là nhỏ nhất.

### Hướng dẫn giải:

Bạn sẽ tính khoảng cách của tất cả các cặp tòa nhà, trừ những tòa nhà nào đã có kết nối rồi thì bạn đặt chi phí kết nối là 0.

Sau khi đã tính xong thì bạn bỏ toàn bộ các cặp kết nối và chi phí đó vào graph rồi chạy thuật toán Prim.

Cây khung nhỏ nhất là kết quả tìm được.

**Độ phức tạp:**  $O(T * N^2 * \log N)$  với  $T$  là số lượng bộ test và  $O(N^2 * \log N)$  là độ phức tạp của thuật toán Prim vì trong bài này ta dùng tất cả cạnh nối giữa các tòa nhà nên số lượng cạnh là  $N^2$ .